NASA Contractor Report 187484, Volume I

A General Multiblock Euler Code for Propulsion Integration,

Volume I: Theory Document

H. C. Chen, T. Y. Su, and T. J. Kao

Boeing Commercial Airplane Group
Seattle, Washington

Contract NAS1-18703
May 1991

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

1. Table of Contents

1. Summary

A general multiblock Euler solver was developed for the analysis of flow fields over geometrically complex configurations either in free air or in a wind tunnel. In this approach, the external space around a complex configuration was divided into a number of topologically simple blocks, so that surface-fitted grids and an efficient flow solution algorithm could be easily applied in each block. The computational grid in each block is generated using a combination of algebraic and elliptic methods. A grid-generation/flow-solver interface program was developed to facilitate the establishment of block-to-block relations and the boundary conditions for each block. The flow solver utilizes a finite-volume formulation and an explicit time-stepping scheme to solve the Euler equations. A multiblock version of the multigrid method was developed to accelerate the convergence of the calculations. The generality of the method was demonstrated through the analysis of two complex configurations at various flow conditions. Results were compared to available test data. Two accompanying volumes, user manuals for the preparation of multi-block grids (Vol. II) and for the Euler flow solver (Vol. III), provide information on input data format and program execution.

2. Introduction

The increases in fuel cost over the last two decades has spurred the development of fuel-efficient propulsive devices for transport aircraft. Studies (Ref. 1, 2) have shown that high speed propfans have the potential of reducing the fuel consumption of conventional subsonic transports. More recently, both fuel efficient advanced turbofan and very-high-bypass superfan engines have been developed for a new generation of advanced transport aircraft.

These propulsion systems offer significant reduction in specific fuel consumption. It is essential, however, to integrate the propulsion system properly with the airframe if the fuel-efficiency advantages are to be realized and optimized in actual flight. NASA Langley Research Center has an on-going research program on the installation of advanced propulsion systems on transport aircraft with supercritical wings (Ref. 3). Complementary to that effort, the present work is concentrated on the development of a computational fluid dynamics (CFD) tool for studying the effects of advanced propulsion systems on airplane lift and drag characteristics.

In developing a CFD tool for engine-airframe integration, one must address the following issues: complex geometry; nonlinear effects associated with transonic flows, rotational flows and flows with non-uniform total pressure and temperature (caused by propfan or turbofan engines); and viscous effects on the configuration surfaces. Ideally, a complex flow such as this can only be resolved with the Navier-Stokes equations. Unfortunately, the Navier-Stokes formulation requires very fine grids to resolve the viscous layers, which, in turn, requires large amounts of computing resources in order to obtain useful results for the analysis of a complex airplane configuration. An alternative is to neglect all viscous terms and reduce the governing equations to the inviscid, Euler equations, so that they can be solved effectively with the available computing resources.

In this report, an Euler-based computational method for the study of engine-airframe integration problems will be presented. Two major tasks were involved in the development of this method; the generation of computational grids; and the development of an appropriate flow solver.

In the area of grid generation, extensive research can be found in the literature. These include the unstructured grid approach (Ref. 4, 5), and the structured grid approach (Ref. 6, 7). The unstructured grid approach is an ideal choice for complex configuration analysis. However, the requirement for large central memory and the relatively low execution efficiency, presently limits the practical applications of the unstructured grid method (Ref. 8-11). The structured grid approach together with a multiblock method to handle complex geometries (Ref. 12-17), offers both geometry flexibility and execution efficiency for large problems. This approach was chosen as the framework for the present study.

The basic philosophy behind the multiblock grid generation process is discussed in Section 3. A grid generation toolkit, program BCON, was developed to facilitate this process. BCON is documented in details in volume II (Ref. 18) of this report.

The Euler-based flow solver was developed for general configuration analysis. Because of the multi-block concept, this General Multi-Block Euler (GMBE) solver is both configuration independent and grid topology independent. The complete flow field, exterior to the configuration, is divided into a number of topologically simple regions or blocks. Each block consists of six faces in the computational plane. Along each face of the block, boundary conditions, together with block-to-block relationship; are specified. Every block has similar data structure, so that the same flow solver algorithm can be used throughout the entire flow field.

Two new features are introduced in the present multiblock flow solver: 1) generalized block- to-block connectivity which allows each block an arbitrary number of neighbors, and 2) general boundary conditions to allow each face an arbitrary combination of boundary conditions. These new features relieve the blocking restrictions encountered in most existing multiblock methods, where each face of the block can only have one neighbor and one type of boundary condition. These features result in a significant reduction in blocks and data communication requirements. As a result, there is lower block boundary overhead and more effective vectorization in the flow solver. These two features can also be very effective for simpler configurations using very few (2-10) blocks (Ref. 19).

Computed results for two test cases are presented to demonstrate the generality of the method. The first case is a wing-mounted propfan configuration in a wind tunnel. The second case is a generic twin-engined, NASA low-wing transport configuration (Ref. 3). Results from both are compared with test data. Volume III (Ref. 20) is the user manual for the Euler solver.

3. Multiblock Grid Generation

The generation of a surface-fitted grid over a geometrically complex configuration, in a manner that is non-configuration specific, is an essential element in the present general flow solver development. In general, a single grid cannot provide adequate topological resolution to cover a complex shape. Consequently, the flow field needs to be divided into a number of simple regions, as illustrated in Figure 1, such that within each region, smooth and well distributed grids with a desirable grid topology can be generated. Each block contains six faces in the computational space. Some faces may have no neighbor, such as the faces on the far-field boundary or on the configuration surface. Others may have multiple neighbors, such as those on the interface boundaries. The configuration surface must lie on block-faces. This is necessary for a non-configuration specific multi-block algorithm. Along each block-face, a combination of boundary conditions may be used. Presently, there is a restriction that the common faces of adjoining blocks have identical grid points.

The multiblock grid generation process is divided into four steps:

1. Blocking of the Complete flow field

The first step in the grid generation procedure is to divide the complete flow field into a number of topologically simpler blocks. Complex geometry, such as a complete airplane, may require many blocks (currently, of the order of 30).

2. Surface Grid Generation on Block Faces

In this step, the grids on each block face must be generated. If the block face is part of a configuration surface, the grid must be interpolated in the parametric space of the original geometry definition to ensure that the grid lies on the configuration surface. All other block faces will be either an interblock boundaries or the far-field boundaries or inlet or exhaust boundaries or propeller disk boundaries. The surface grid for these faces may be generated either by an algebraic method or by an elliptic grid generation method.

3. Volume-Grid Generation

After the surface grids of each block have been properly prepared, the EAGLE (Ref. 9) grid generation package can be used for field-grid generation. This package uses both algebraic/transfinite interpolation and elliptic grid-generation methods for volume-grid generation.

4. Block-to-Block Relation and Boundary Condition Definition

The block-to-block relation file is required by the flow solver. This file establishes the flow field communications between adjacent blocks. In addition, a block-boundary condition file in needed by the flow solver in order to apply the proper boundary conditions along each block face. The block-to-block relations are established such that every common block boundary is assigned a unique record number, together with the two dimensional grid index associated with that boundary. The record number of its direct neighbors are also identified. This information enables the Euler solver to access appropriate data from neighboring blocks for the convection and dissipation flux calculations. The block-boundary condition is assigned in patches. Each patch represents a unique boundary condition with a given 2D index. Each block face can have any combination of the following types of boundary conditions: 1) interface, 2) solid surface, 3) inlet, 4) exhaust, 5) far-field 6) centerline, 7) mirror image, and 8) actuator disk.

BCON is used to prepare the geometry data and job deck inputs to run the EAGLE code (Ref. 6) for volume grid generation. BCON will also generate the block-to-block relation and block-boundary condition files for the Euler solver. In addition, BCON can be useful for data checking to ensure that all information is correct, before it is passed to the flow solver.

4. General Euler Solver Algorithm

The Euler solver was derived from Jameson's cell-centered finite-volume approach (Ref. 21) where the unsteady Euler equations are solved using a multi-stage Runge-Kutta time-integration method; see Appendix A. The main objective of this work was to develop a flow solver that can accept any combination of grid topologies for the analysis of complex flows over arbitrary configurations. Jameson's solver uses a C-grid-topology which is adequate for the analysis of wing/body flow. To achieve the objective of treating arbitrary geometries, the flow solver must be non-configuration specific and non-grid-topology specific. This objective required the adaptation of Jameson's flow solver to a multiblock grid. In the multiblock version of the Euler flow solver, boundary condition surfaces are restricted to the block faces (Sec. 3). Then the interior of every block would contain only regular flow field grid points, such that the flow field of each block can be solved using the same flow solver. Requirements for the adaptation of Jameson's flow solver to multiblock grid include: development of a multiblock-compatible version of the multigrid method; algorithm improvement for treating irregular and singular grid points in the flow field; and implementation of boundary conditions in the multiblock environment. A flow solver that addressing all these issues is necessarily demanding on computing resources. Thus efficient management of computer memory is an important aspect of GMBE.

The multigrid strategy employed in the present study is a simple V-cycle multigrid applied within each individual block of the flow field. Two main reasons for using V-cycle multigrid in a block-by-block manner are: 1) to minimize data transfer from main memory to the secondary memory (Ref. 22) and 2) to significantly reduce the programming complexity for the present multiblock solver. However, this restricts the multigrid calculation to one block at a time. The residual collection and the correction interpolation at the grid cell near the block face can not be done in the same way as the single block approach. This can affect the overall convergence. It may also cause numerical oscillations for the analysis of supercritical wings with rooftop-type pressure distributions (Ref. 12). A simple way to circumvent this problem is to turn-off the multigrid calculation after the solution converges to a certain level of accuracy. Iterative computations will then be continued in the single grid mode until desired convergence is achieved. In dealing with complex geometries such as the wing/body/under-wing nacelle/pylon configuration shown in Fig. 2, it can be very time-consuming to ensure that every block of a multiblock grid does not contain an odd number of cells in some given direction within the computational space. An adaptive multigrid strategy has been developed which recognizes blocks which have an odd number of grid cells in any given direction. A block containing an odd number of grid cells, in any direction, will be solved by single-grid calculations. Different multigrid levels in other blocks are employed based on the local grid cell distributions. In order for the adaptive multigrid strategy to be effective, the non-multigrid blocks should be small in number and size; most of the flow field should be solved using multigrid.

Jameson's cell-centered finite-volume scheme (Ref. 21) in the present Euler solver is accurate and reliable provided that the grids in the flow field are smooth and well distributed. However, irregular and singular grids in the flow field can affect the accuracy and the reliability. Examples are the collapsed edge in front of the wing leading edge near the wing tip (Fig. 3a), and the fictitious corner along the fan cowl surface (Fig. 3b). Special techniques must be applied in these regions in order to obtain an accurate and reliable solution. There are two main concerns here: 1) to ensure flux balance; and 2) to minimize the sensitivity of the dissipation terms to grid irregularities such as high aspect ratio cells and fictitious corners. The convective flux and the dissipative flux along these grid lines must be fully conservative. Flux terms along each face of the cell are computed and used consistently for the flux balance of the two cells adjacent to the face. Treatment of zero volume cells near the collapsed edge is given in Appendix A. Another improvement that has been added to the flow solver is the new dissipation formulation based on a spectral radius scaling (refs. 23-25). The new formulation improves the consistency of the solution by decreasing sensitivity to the dissipation parameters. The improvement over conventional dissipation formulations (Ref. 26) is especially pronounced on grids with a large aspect ratio or with nonsmooth mesh distributions (Ref. 27). Further details can be found in Appendix B.

On each face, the calculation process is dependent upon the type of boundary condition specified. If it is an interblock boundary surface, conservation of mass, momentum, energy, and dissipation flux are ensured across the block boundary. The fourth-order dissipation requires two layers of data from every adjacent block in order to formulate a flux term that is identical to the one used in single-block calculations.

Characteristics boundary conditions (Ref. 28) have been implemented along the far-field boundaries. This ensures proper propagation of information in and out of the boundaries of the flow domain. This is crucial for numerical accuracy and stability, especially for cases, involving powered nacelles or propellers, with non-uniform total temperature. The tangency boundary condition is implemented on the configuration surface, where the normal momentum equation is used to compute the surface pressure.

Propeller power loading is simulated by an actuator disk where the distributions of total pressure, total temperature, and swirl are prescribed. Another option for simulating propeller power effects is to prescribe the thrust, normal force, and side force on the propeller disk and the work added to the flow by the propeller. The flow variables downstream of the disk are related to their upstream values through the continuity, momentum, and energy equations. The major advantage of this method is that the effects of angle of attack, as well as the influence of side flow, can easily be simulated through the input of normal force and side force distributions on the propeller disk. This method is described in more detail in References 29 and 30.

Boundary condition methods for turbofan engine inlet and exhaust are given in Reference 31. The nacelle center line, as well as any grid line in the flow field representing a collapsed grid plane, is treated as solid boundary with zero face area. Also sufficient ghost cells are employed in the mirror-image boundary conditions for producing exact symmetry solutions. Further details on boundary condition implementations are given in Appendix C.

The efficient management of directly addressable central memory is a necessity in large scale CFD simulation because central memory is a scarce resource in a multi-processor supercomputer. Therefore, for a general flow field simulation, it would be very useful to adjust the memory requirement to each grid for a specific CFD application. To facilitate this process, a pre-processor to the general multiblock Euler code was written to manage the central memory requirement for the Euler calculations. Grid information is read in and processed by this pre-processor code to ensure optimum block-storage area in the memory for storing the flow field and geometry information of each block. This information includes the flow solution vector, grid coordinator vector and cell-volume vector. The information is then moved to a working memory location, in a block-by-block manner, to allow the flow solver to update the solution. The updated solution is then stored back in the block-storage area. The working memory is sufficiently large to hold the data for any one block including dissipation vectors, and other temporary arrays required for the Euler code multigrid calculations.

5. Results and Discussions of Test Cases

5.1 Case 1: Wing-Mounted Propfan in Wind Tunnel

A wing-mounted propfan configuration in a wind tunnel, Fig. 4, was analyzed at $M_\infty = 0.167$, $\alpha = 0.0$ deg. Solid wind tunnel wall boundary conditions are included in the calculation to account for blockage effects. The propeller power input (in terms of total pressure, total temperature, and swirl) is prescribed on the propeller disk for a low power setting. As the general multiblock Euler code is neither configuration nor grid-topology specific, it allows for the selection of a grid system that is optimum for a specific application. For this test case, a cylindrical type of grid is used to provide good grid resolution on the propeller disk and nacelle surface. Fig. 5 shows the configuration and several slices of the grid used in the analysis. The grid slices in Fig. 5c and 5d show the dense grid near the solid body boundary on the nacelle and wing, as well as near the tip of the propeller disk. The complete flow field is divided into two zones by a cylindrical-type grid surface. At the propeller disk station, the cross-section of this surface coincides with the tip of the propeller disk. The inner grid zone between this surface and the configuration surface is divided into nine blocks. The outer grid zone is also divided into nine blocks. The field grid consists of 860,000 points.

The isobars on the configuration surface for the power-on case are shown in Fig. 6. The effects of propeller power are depicted in Fig. 7 by comparing the propeller power-on (solid line) with power-off (dashed line) results. Wing section lift has been decreased on the downwash side due to the effects of swirl, Fig. 7(a). This is accompanied by an increase of section lift on the upwash side, Fig. 7(b). The computed section pressure coefficient (C_p) distributions showed excellent agreement with test data for both the downwash, Fig. 8(a), and upwash Fig. 8(b), locations.

5.2 Case 2: Low Wing Turbofan Transport

To further demonstrate the code's capability to handle complex geometries, a NASA low wing transport configuration consisting of wing/body/under-wing nacelle/pylon was analyzed. This model was tested in the NASA Langley 16-foot Transonic Tunnel (Ref. 3) for the investigation of nacelle/pylon installation on wing/body. Detailed information about the model can be found in Ref. 3. The surface grid is given in Fig. 2. The entire flow field consisted of 29 blocks with a total of 1,140,000 grid points. The geometric complexity required the use of a composite grid system for proper resolution of the flow field around the engine/nacelle and also near the wing. The composite grid system included an H-type grid for the external flow field and a cylindrical grid for the nacelle inlet and exhaust flows (Fig. 9). To allow for the study of engine power effects, the detailed gridding around the nacelle was constructed with a grid plane on the inlet fan-face, a grid plane on the exhaust face of the primary (or core) exhaust flow and another grid plane on the exhaust face of the fan exhaust flow. To simulate a flow-through condition, the total pressure and total temperature at both exhaust planes were set at freestream values. The velocity at the fan inlet face was arbitrarily set at 0.6 times the freestream velocity to account for inlet blockage. Fig. 10 shows the isobars on the configuration surface for the flow-through case when $M_\infty = 0.77$ and $\alpha = 0.5$ degree. It can be seen that detailed flow resolutions in the nacelle-pylon region are obtained with the present multiblock Euler solver. Figure 11 compares the computed wing surface pressures at two span stations inboard of the nacelle with test data. The differences could have arisen because the viscous effects and the aeroelastic effects were not accounted for in the present analysis.

This configuration was also analyzed with engine power-on to demonstrate the code's capability to simulate power-on conditions. The total pressure P_0 (normalized by the freestream static pressure) for the core exit was set at 1.92 and at 1.60 for the fan exit. The inlet velocity at the fan face was set at 0.6 times the freestream velocity. Figure 12 compares the C_p distribution, for the engine power-on and power-off cases, along the outboard intersection line between the pylon and the primary cowl. The difference in the pylon surface pressure distributions are discernable even at this low power setting.

Finally, the same configuration was analyzed at an off-design condition (M_∞= 0.8, and α = 0.474 deg). The results are compared with test data in Fig. 13. The discrepancy between experimental and computed shock locations are more pronounced because of the stronger viscous effects at this higher transonic Mach number.

6. Concluding Remarks

A grid generation and block-to-block boundary condition pre-processor (BCON) and a general multiblock Euler (GMBE) solver were developed. The generality of the program is demonstrated through the analysis of two complex configurations at different flow conditions. The analysis results agree well with test data provided that the viscous effects and aeroelastic effects are not predominant. When used within these conditions, the code is a useful and effective analysis tool for engine/airframe integration studies. Accompanying volumes are user manuals for BCON (Vol. II) and GMBE (Vol. III)

7. References

1. Boeing Commercial Airplane Co., "Energy Consumption Characteristics of Transports Using Prop-Fan Concept," NASA CR-137937, NASA, Moffett Field, CA., October 1976.

2. Carlson, J. R., and Pendergraft, Jr., O. C., "An Experimental Investigation of an Advanced Turboprop Installation on a Swept Wing at Subsonic and Transonic Speeds," NASA Langley Research Center, NASA TP-2729, September, 1987

3. Pendergraft, Jr., O. C., Ingraldi, A. M., Re, R. J., and Kariya, T. T., "Nacelle/Pylon Interference Study on a 1/17-Scale, Twin-Engine, Low-Wing Transport Model,", AIAA 89-2480, July 1989.

4. Peraire, J, Morgan, K., and Peiro, J., "Unstructured Finite Element Mesh Generation and Adaptive Procedures for CFD," AGARD Conference Proceeding No. 464, Applications of Mesh Generation to Complex 3-D Configurations, May 1989.

5. Baker, T. J., "Unstructured Mesh Generation By a Generalized Delauny Algorithm Generation and Adaptive Procedures for CFD," AGARD Conference Proceeding No. 464, Applications of Mesh Generation to Complex 3-D Configurations, May 1989.

6. Lijewski, E., and Cipolla, J., "Program EAGLE User's Manual," USAF Armament Laboratory Technical Report, AFATL-TR-88-117, Eglin AFB, September 1988.

7. Steinbrenner, J. P., Chawner, J. R., and Fouts, C. L., "Multi-Block Grid Generation in a Interactive Environment," AIAA 90-1062, June 1990.

8. Mavriplis, D. J., "Euler and Navier-Stokes Computations for Two-Dimensional Geometries Using Unstructured Meshes," NASA CR-181977, ICASE Report No. 90-03, January 1990.

9. Jameson, A. and Baker, T. J., "Improvement to the Aircraft Euler Method ," AIAA 87-0452, January 1987.

10. Peraire, J., Peiro, J., Formaggia, L., and Morgan, K., "Adaptive Numerical Solutions of the Euler Equations in 3D Using Finite Elements," presented in 11th ICNMF, June 1988, published in Lecture Notes in Physics, Vol. 323, Springer-Verlag, 1989.

11. Frink,, N. T., Parikh, P., and Pizadeh, S., "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes," AIAA 91-0102, January 1991.

12. Yu, N. J., Chen, H. C., Su, T. Y., and Kao, T. J., "Development of a General Multiblock Flow Solver for Complex Configurations," Proceeding of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics, October 1989.

13. Fritz, W., Haase, W., and Seibert, W., "Mesh Generation for Industrial Application of Euler and Navier Stokes Solvers," Conference on Automated Mesh Generation and Adaption, Grenoble, France, October 1987.

14. Amendola,A., Tognaccini, R., Boerstoel, J. W., and Kassies, A., "Validation of A Multi-Block Euler Flow Solver with Propeller-Slipstream Flows," AGARD Fluid Dynamics Panel Symposium, Lisbon, 1988.

15. Raj, P., Olling, C. R., Sikora, J. S, Keen, J. M., Singer, S. W., and Brennan, J. E., "Three-Dimensional Euler/Navier-Stokes Aerodynamic Method (TEAM) Volumes 1-3," AFWAL-TR-87-3074, June 1989.

16. Sawada, K., and Takanashi, S., "A Numerical Investigation on Wing/Nacelle Interferences of USB Configurations," AIAA 87-0455, January 1987.

17. Flores, J., Holst, T. L., Kaynak, U., Gundy, K., and Thomas, S. D., "Transonic Navier Stokes Wing Solution Using a Zonal Approach, Part 1, Solution Methodology and Code Validation," AGARD 58th Fluid Dynamics Panel Symposium, Aix-en-Provence, France, April 1986.

18. Su, T. Y., Appleby, R. A., and Chen, H. C., "A General Multiblock Euler Code for Propulsion Integration, Volume II: User Guide for BCON, Pre-Processor for Grid Generation and GMBE, " NASA CR-187484, Volume II, May 1991.

19. Eriksson, L., and Orbekk, E., "Algebraic Block-Structured Grid Generation Based on a Macro-Block Concept," AGARD Conference Proceeding No. 464, Applications of Mesh Generation to Complex 3-D Configurations, May 1989.

20. Chen, H. C., "A General Multiblock Euler Code for Propulsion Integration, Volume III: User Guide for the Euler Code," NASA CR-187484, Volume III, May 1991.

21. Jameson, A., and Baker, T. J., "Solution of the Euler Equations for Complex Configurations," AIAA 83-1929, June 1983.

22. Rossow, C.-C., "Efficient Computation of Inviscid Flow Fields Around Complex Configurations Using a Multi-Block Multigrid Method," Fifth Copper Mountain Conference on Multigrid Methods, March 1991.

23. Jameson, A., "Successes and Challenges in Computational Aerodynamics," AIAA 87-1184, June 1987.

24. Swanson, R. C. and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations," AIAA 87-1107, June 1987.

25. Martinelli, L., "Calculation of Viscous Flows with a Multigrid Method," Ph. D. Thesis, Department of Mechanical and Aerospace Enginnering, Princeton University, October 1987.

26. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Scheme," AIAA 81-1259, June 1981.

27. Chen, H. C., and Yu, N. J., "Development of a Highly Efficient and Accurate 3D Euler Flow Solver," presented in 11th ICNMF, June 1988, published in Lecture Notes in Physics, Vol. 323, Springer-Verlag, 1989.

28. Allmaras, S. R., "A Coupled Euler/Navier-Stokes Algorithm for 2-D Unsteady Transonic Shock/Boundary-Layer Interaction," PhD Thesis, MIT, February 1989.

29. Yu, N. J., Samant, S. S. and Rubbert, P. E., "Flow Prediction for Propfan Configuration Using Euler Equations," AIAA 84-1645, June 1984.

30. Yu, N. J., and Chen, H. C., "Flow Simulations for Nacelle-Propeller Configurations Using Euler Equations," AIAA 84-2143, August 1984.

31. Chen, H. C., Yu, N. J. Rubbert, P. E., and Jameson A., "Flow Simulations for General Nacelle Configurations Using Euler Equations," AIAA 83-0539, January 1983.

32. Wigton., L. B., and Swanson, R. C., "Variable Coefficient Implicit Residual Smoothing," 12th International Conference on Numerical Methods in Fluid Dynamics, Oxford, England, July 1990.

8. Appendices

Appendix A

Numerical Method for the Euler Solver

The basic numerical scheme follows Jameson's finite volume formulation (Ref. 26). In this formulation, the computational domain is divided into a large number of computation cells each treated as a control volume for the flux balance of the governing conservation laws. For the Euler equations

$$\partial w/\partial t + \partial f/\partial x + \partial g/\partial y + \partial h/\partial z = 0 \qquad \text{(A1)}$$

$$w = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \quad f = \begin{bmatrix} \rho u \\ p+\rho u^2 \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix} \quad g = \begin{bmatrix} \rho v \\ \rho vu \\ p+\rho v^2 \\ \rho vw \\ \rho vH \end{bmatrix} \quad h = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ p+\rho w^2 \\ \rho wH \end{bmatrix}$$

Here, the conventional notations ρ, u, v, w, p, E are used for density, three components of velocity, pressure, and the total energy. The total enthalpy H is defined as

$$H = E + p/\rho = \gamma p/(\gamma -1)\rho + (u^2 + v^2 + w^2)/2 \qquad \text{(A2)}$$

with γ representing the ratio of specific heat. Equations (A1, A2) are augmented by the perfect gas law as the thermal equation of state, and the caloric equation of state for the perfect gas. In the governing equations, the density and the pressure are normalized with respect to their freestream values. The total energy and the total enthalpy are normalized with respect to the ratio between the freestream pressure and freestream density p_∞/ρ_∞, and the velocity is normalized with respect to the square root of p_∞/ρ_∞.

The unsteady Euler equations are solved by the multistage Runge-Kutta time-stepping scheme which for an n stage may be written (Ref. 23) for the time step n to n+1 as:

$$\begin{aligned}
&w^{(0)} = w^{n} \\
&w^{(1)} = w^{(0)} - \alpha_1 \Delta t R^{(0)} \\
&\text{- - - - - - - - - - - - - - -} \qquad\qquad \text{(A3)} \\
&w^{(m-1)} = w^{(0)} - \alpha_{m-1} \Delta t R^{(m-2)} \\
&w^{n+1} = w^{(m)} = w^{(0)} - \alpha_{m} \Delta t R^{(m-1)} \\
&\text{with } R^{(m)} = C(w)^{(m)} + D(w)^{(m)}
\end{aligned}$$

Here, $C(w)^{(m)}$ represents convective flux terms, and $D(w)^{(m)}$ represents artificial dissipation terms. The convective terms {f}, {g} and {h} from Equation (A1) are discretized as given in Reference 26. The dissipation terms are added to control numerical oscillations and to capture the shocks. Details on the dissipation formulation can be found in Appendix B. Central difference approximations are used for both convective and dissipative term calculations.

The implicit residual smoothing (IRS) method of Ref. 32 was employed to extend the stability range of the Runge-Kutta explicit time marching scheme. IRS also significantly enhances the smoothing characteristics of the time marching scheme for use with multigrid. This version of IRS provides superior performance, particularly for very fine grid calculations.

In addition, for flow with uniform enthalpy, enthalpy damping procedure (Ref. 26) is used to accelerate the convergence of Euler solutions. It modifies the Euler equations by adding forcing terms proportional to the difference between the local and free stream values of total enthalpy. The formulation of spatial discretization ensures that constant total enthalpy condition is consistent with the steady-state solution of the difference equations. In so doing, the steady-state solution is not altered by these additional terms.

Many grid surfaces converged into a collapsed edge as shown in Fig. 3a. Extending of such coinciding surfaces beyond the collapsed edge would result in zero volume cells. These cells must be eliminated from the block grid definition because flow physics requires that the adjacent blocks on both sides of the zero thickness surface access information from each other directly. This was acheived by specifying a special relationship between the adjacent blocks. The information is given in the block-to-block relationship file that was created by BCON (Vol. II) during the process of multiblock grid generation.

Appendix B

Artificial Dissipation

The artificial dissipation model used here is based on the model devised by Jameson and Baker (Ref. 21) for 3-D Euler equations. The original model was modified for scaling in the coordinate directions as suggested by Martinalli (Ref. 25) and Swanson and Turkel (Ref. 24). These concepts are outlined below.

A spectral radius scaling on the dissipation terms is used to minimize the amount of artificial dissipations. The dissipation term D(w) is expressed as

$$D(w) = D(w)_i + D(w)_j + D(w)_k \tag{B1}$$

where the indices i, j, and k denote contributions from the ξ, η, and ζ directions respectively. The formulas for the contribution from ξ-direction are discussed here. The corresponding formulations in the two other directions can be obtained through appropriate permutation of the indices.

$$D(w)_i = d_{i+1/2,j,k} - d_{i-1/2,j,k} \tag{B2}$$

These dissipation terms are a blend of second and fourth order differences with coefficients adapted to the flow.

$$d_{i+1/2,j,k} = Bi_{i+1/2,j,k}(\varepsilon^{(2)} - \varepsilon^{(4)}\delta_\xi{}^2)(W_{i+1,j,k} - W_{i,j,k}) \tag{B3}$$

where the scaling factor $Bi_{i+1/2,j,k} = \lambda i_{i+1/2,j,k}$ is the spectral radius in the ξ-

direction at face (i+1/2, j, k). Bi is first computed at cell center (i, j, k)

$$Bi_{i,j,k} = \lambda i_{i,j,k} = (q_{n_{i,j,k}} + a_{i,j,k})(S_{i-1/2,j,k} + S_{i+1/2,j,k})/2 \tag{B4}$$

where $a_{i,j,k}$ is the speed of sound at the cell center and $S_{i+1/2,j,k}$ is the cell face area. Bi at the cell face (i+1/2, j, k) is then computed by averaging the two values at the adjacent cell centers, i. e.,

$$Bi_{i+1/2,j,k} = (Bi_{i,j,k} + Bi_{i+1,j,k})/2 \tag{B5}$$

In Equation (B3) $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are the adaptive coefficients and the $\delta_\xi{}^2$ is the second difference operator in the ξ-direction. The coefficient $\varepsilon^{(4)}$ controls fourth order dissipation which provides the background dissipation in the smooth part of the flow to suppress numerical oscillations. Near shocks it is reduced to zero. The coefficient $\varepsilon^{(2)}$ controls second order dissipation which is used to capture the shocks. The term $\varepsilon^{(2)}$ is made proportional to the normalized second difference in the pressure

$$\varepsilon^{(2)} = K^{(2)} \max(\nu i_{i,j,k}, \nu i_{j+1,j,k}) , \tag{B6}$$

$$\nu i_{i,j,k} = |(p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k})/(p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k})| , \tag{B7}$$

and

$$\varepsilon^{(4)} = \max[0, (K^{(4)} - \varepsilon^{(2)})] , \tag{B8}$$

where typical values of the constant $K^{(2)}$ and $K^{(4)}$ are 1/4 and 1/256, respectively.

This isotropic dissipation formulation leads to good shock capturing and is adequate for computational grids of moderately high aspect ratio (of the order 10). However, for grids containing very high aspect ratio grid cells (e. g., 500), the dissipation coefficients may become imbalanced in different coordinate directions. In these cases, the artificial dissipation should be anisotropic in the three coordinate directions. According to the modification developed by Martinelli (Ref. 25), the scaling factors of Equation (B4) are adjusted as follows:

$$Bi_{i,j,k} = \Phi(ri)\lambda i_{i,j,k} \tag{B9}$$

where

$$ri = (\lambda j_{i,j,k} + \lambda k_{i,j,k})/\lambda i_{i,j,k}$$

and

$$\Phi(ri) = 1 + ri^{\alpha} \quad \text{where} \quad 0 < \alpha < 1$$

$\alpha = 2/3$ is recommended by Martinelli.

The pressure sensor term based on the normalized second difference of pressure in Equation (B7) has some interesting characteristics. Across a shock wave, this term will be large since a large pressure variation can lead to large second pressure difference. In the smooth part of the flow, this term will be insignificant. For some low speed flow calculations where the shock wave does not occur, this pressure sensing term may become significant near the leading edge, where the flow is undergoing rapid expansion. For this type of flow, the pressure sensor of Equation (B7) is replaced by a sensor using an entropy function f defined as

$$f = (P/\rho)^{\gamma}$$

where

$$\nu i_{i,j,k} = |(f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k})/(f_{i+1,j,k} + 2f_{i,j,k} + f_{i-1,j,k})| \qquad \text{(B9)}$$

The value of this sensor term will be small in regions of flow where the entropy function f (or entropy) is uniform or a slow varying function.

In the coarse grid a simplified dissipation model is used by replacing Equation (B3) with

$$d_{i+1/2,j,k} = K^{(0)} Bi_{i+1/2,j,k}(W_{i+1,j,k} - W_{i,j,k}) \qquad \text{(B10)}$$

where typical value of the constant $K^{(0)}$ is 1/128.

Appendix C

Boundary Condition Implementations

The boundary conditions discussed in this Appendix include: 1) Solid-surface boundary condition, 2) Far-field boundary condition, 3) Nacelle-inlet boundary condition, and 4) nacelle exhaust plane/propeller disk boundary conditions.

1). Solid-surface boundary condition

Along the configuration surface, the tangency condition is enforced by setting $\bar{q} \cdot \bar{n} = 0$ where $\bar{q}$ is the velocity vector and n is the unit outward normal vector. For the cell-centered scheme used in the present work, the pressure on the configuration is also needed to evaluate the convective flux terms in the momentum and energy equations. Simple extrapolation from adjacent cell centers (see Vol. III pages 8 and 16) can be used. Optionally, to improve numerical accuracy, a normal momentum equation that relates the normal derivative of pressure to the streamwise and spanwise derivatives can be used along a solid surface in a plane of constant j the relation becomes

$$S_{2j} S_{ij}\ \partial p/\partial \xi_i - \rho U_i U_j \partial(S_{2j})/\partial \xi_i = 0 \qquad \text{(C1)}$$

with $S_{ij} = h\ \partial \xi_i/\partial x_j$, the projected area of cell face-i in the x_j direction, where $x_1 = x$, $x_2 = y$, $x_3 = z$, and $\xi_1 = \xi$, $\xi_2 = \eta$, $\xi_3 = \zeta$, $h = |\partial x_i/\partial \xi_j|$, the cell volume, and $U_i = S_{ij} u_j$, the contravariant velocity component in the ξ_i-direction.

2). Far-field boundary condition

Because the calculation is performed in a finite region, correct implementation of the far-field boundary condition is essential for numerical accuracy and stability.

The characteristic boundary condition based on locally one-dimensional Euler equations is used. Along each face of the far-field boundary cell, the velocity component normal to the far-field boundary is computed and checked.

2(a) For a subsonic freestream condition, if it is an inflow boundary, there are four incoming characteristics requiring the specification of four quantities: the tangential velocity; the enthalpy; the entropy; and one Riemann variable evaluated from upstream values. This information together with another Riemann variable extracted from the interior flow field are used to determine all flow variables, i.e.,

$$\begin{aligned} &q_t = q_{t_\infty}, \quad s = s_\infty, \quad H = H_\infty , \\ &q_n - 2a/(\gamma - 1) = q_{n_e} - 2a_e/(\gamma - 1) , \qquad \text{(C2)} \\ &q_n + 2a/(\gamma - 1) = q_{n_\infty} + 2a_\infty/(\gamma - 1) . \end{aligned}$$

where s denotes the entropy, a denotes the sonic speed, q the velocity, subscript e representing values extrapolated from the interior cells adjacent to the boundary, subscript ∞ denoting freestream values, subscript t implying tangential, and subscript n implying normal components.

2(b) For the subsonic outflow boundary, specification of one quantity is required for the incoming characteristic. One Riemann variable based on freestream values together with four extrapolated quantities are used to calculate the flow variables, i.e.,

$$\begin{aligned} &q_t = q_{t_e}, \quad s = s_e, \quad H = H_e, \\ &q_n - 2a/(\gamma - 1) = q_{n_\infty} - 2a_\infty/(\gamma - 1) , \qquad \text{(C3)} \\ &q_n + 2a/(\gamma - 1) = q_{n_e} + 2a_e/(\gamma - 1) . \end{aligned}$$

This approximation assumes a uniform total temperature distributions down stream in the far-field. This assumption is incorrect for powered nacelle. Therefore, for powered nacelles an alternative downstream far-field boundary condition is specified. The conservative variables ρ, ρu, ρv, ρw, and ρE are extrapolated from their upstream values, and the pressure is set to the freestream value.

2(c) For a supersonic inflow boundary, all characteristics are incoming. Hence all variables are assigned to their freestream values.

2(d) For a supersonic outflow boundary, there are no incoming characteristics. Hence, all flow variables are extrapolated from the interior flow solutions.

3). Nacelle-inlet boundary conditions

These boundary conditions are described in more detail in Ref. 31. The nacelle inlet face is treated as an outflow boundary. This is possible because the interior of the nacelle is not modeled. Three choices of boundary conditions exist:

a). Mass flux boundary condition

If the capture stream tube area of the nacelle, i.e., A_∞ is specified, the mass flux at the inlet face is given by

$$(\rho q_n)_f = (\rho_\infty A_\infty q_\infty)/A_f \tag{C4}$$

with A_f representing inlet face area. This is equivalent to specifying an inflow area ratio.

b). Velocity boundary condition

For a given captured stream tube area A_∞ , the mass flux boundary condition can be converted into a velocity boundary condition using the isentropic relation to relate density to the local velocity, i.e.,

$$q_n/q_\infty = (A_\infty/A_f)[1 - (\gamma-1)(M_\infty)^2\{(q_n/q_\infty)^2 - 1\}/2]^{-1/(\gamma-1)} \qquad \text{(C5)}$$

A simple Newton's iteration procedure is written to solve the above nonlinear equation for q_n. When shock waves are presented in the inlet, the accuracy of this equation deteriorates as the shock waves get stronger. A well designed inlet for subsonic transport, however, should not have strong shock waves in the inlet. Test Case 2 in Section 5.2, using $q_n = 0.6q_\infty$, is equivalent to using an inflow area ratio $A_\infty/A_f = 0.66$.

c). Pressure boundary condition

The static pressure p along the inlet face can be prescribed. The rest of the flow variables are extrapolated from the upstream flow field.

4). Nacelle exhaust plane/propeller disk boundary conditions

The nacelle exhaust plane as well as the downstream side of the propeller disk can have similar boundary conditions. In either case, a subsonic inflow boundary is assumed.

4(a) The first type of boundary condition that can be applied to both nacelle exhaust and propeller disk prescribes the total pressure, total temperature and the swirl on boundary. These quantities are then used to evaluate the Mach number, static pressure and density according to the following formulae

$$M = [2q^2/(\gamma-1)\{2\gamma(T_o)^2/(\gamma-1) - q^2\}]^{1/2},$$

$$p = p_o/[1 + (\gamma-1)M^2/2]^{-\gamma/(\gamma-1)}, \qquad \text{(C6)}$$

$$\rho = \gamma p/[(\gamma - 1)(C_p T_o - q^2/2)]$$

C_p is the specific heat at constant pressure. The subscript o refers to "total" quantities. The flow speed q is obtained by extrapolation from the latest iterative value from downstream. The swirl distributions are then used to give the u, v, and w velocity components from q.

4(b) Another type of boundary condition frequently used for propeller simulation prescribes the distribution of the thrust per unit area (F_x), the normal force per unit area (F_y), the side force per unit area (F_z), and the work done by the propeller (Q) to calculate the flow variables downstream of the propeller disk. These quantities along with the velocity extrapolated from the flow field are used in the momentum equations to solve for ρ, u, v, w and p, downstream of the disk.

$$\begin{aligned}
\rho U u + S_x p &= \rho_1 U_1 u_1 + S_x p_1 + SF_x \,, \\
\rho U v + S_y p &= \rho_1 U_1 v_1 + S_y p_1 + SF_y \,, \\
\rho U w + S_z p &= \rho_1 U_1 w_1 + S_z p_1 + SF_z \,, \qquad \text{(C7)} \\
u^2 + v^2 + w^2 &= q^2 \,, \\
\rho &= \gamma p/[(\gamma - 1)(C_p T_{o_1} + SQ/\rho U - q^2/2)]
\end{aligned}$$

where U denotes the contravariant velocity component in the ξ-direction with subscript 1 denoting upstream conditions. S_x, S_y, and S_z are the projected areas in the Cartesian directions of a cell face on the propeller disk. S is the cell face area.

In either type of propeller disk boundary condition method, the boundary condition on the upstream side of the propeller disk is obtained by setting the mass flux to be equal to the mass flux on the downstream side. Here, the subsonic inflow assumption implies that both the upstream Mach number M_1 and the downstream Mach number M_2 are less than unity. In practice (Ref. 30) transient supersonic flow could sometimes appear at the upstream side of the disk during the early cycles of the iteration process. When M_1 is greater than one, no boundary condition can be assigned along the upstream side of the disk. The flow variables needed for the calculations are obtained from upstream extrapolation. The resulting normal mass flux is then used as a boundary condition for the downstream side of the disk. For a subsonic downstream flow ($M_2 < 1$) this implies that one boundary condition described in either type of method must be deleted in order to have a well posed boundary-value problem. In the case of Method 4(a), the total temperature T_0 is deleted, whereas in the case of Method 4(b), the work term Q is deleted. In this way, the primary influence of the propeller (the thrust) can still be properly simulated. For test cases that are representative of candidate installations, such transient supersonic flow no longer exists during the latter stage of iteration process, and the proper boundary conditions discussed in Methods 4(a) and 4(b) are reinstated. The present propeller theory is not valid for the case where both M_1 and M_2 are greater than one.

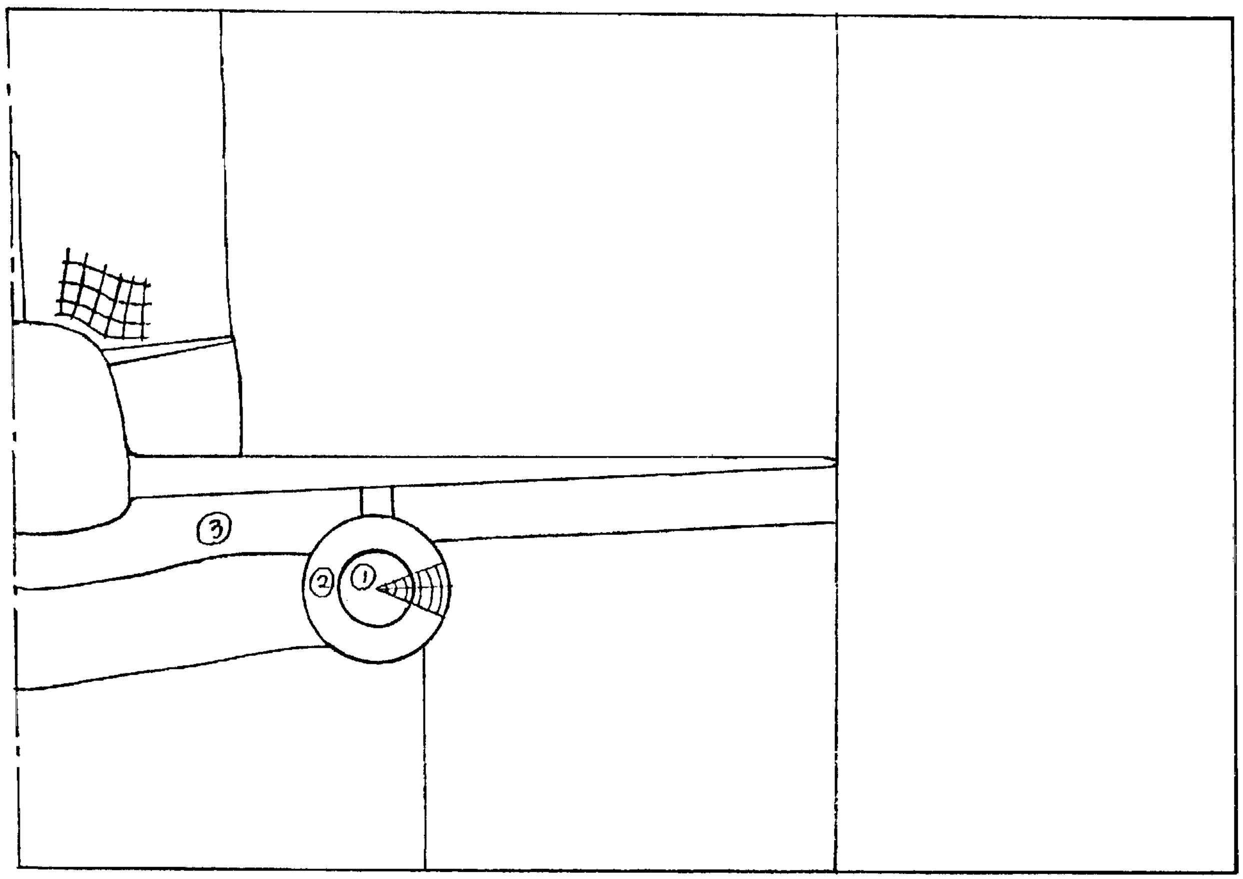
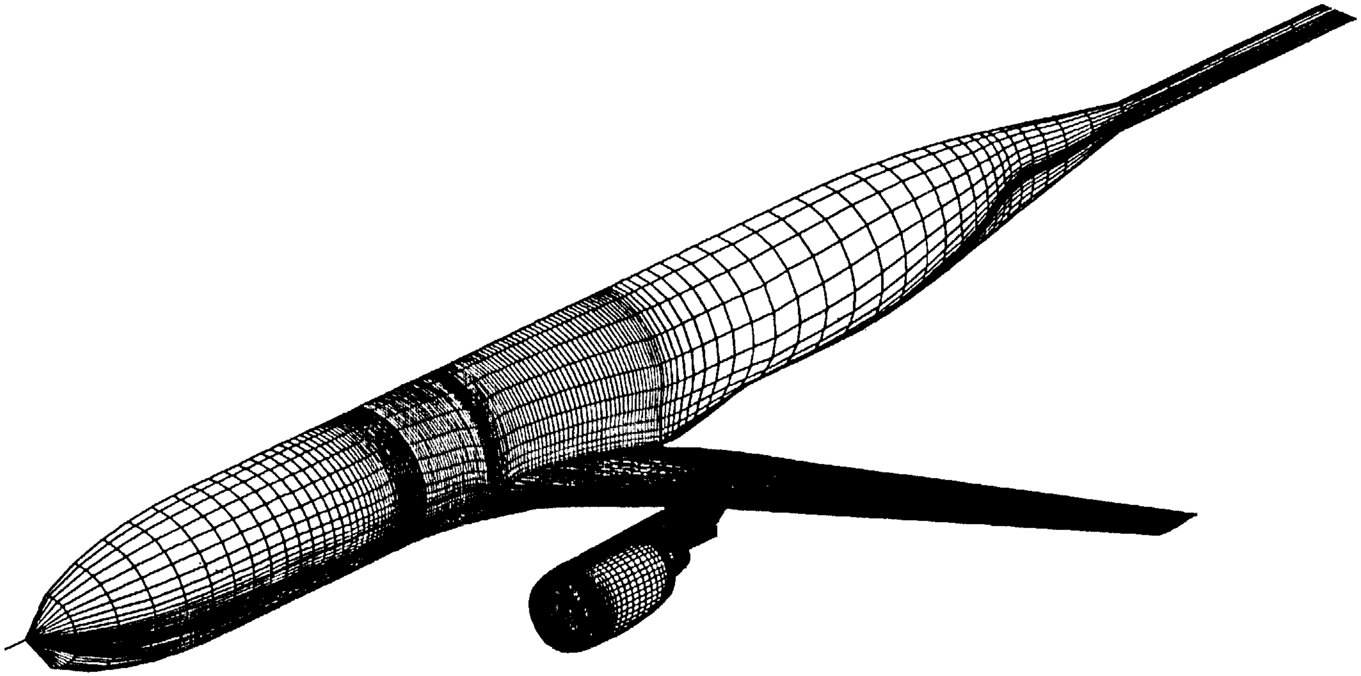
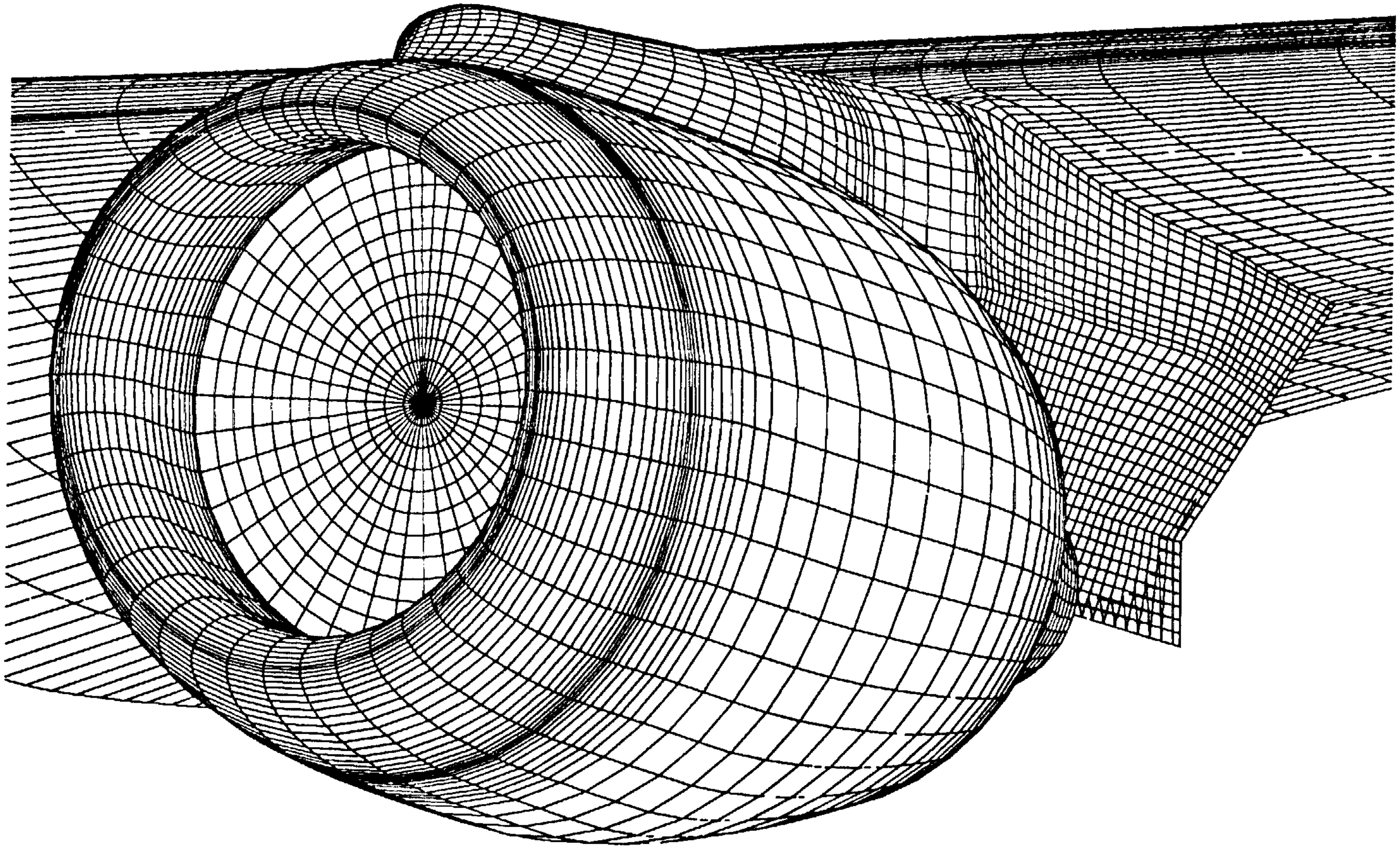


Fig. 1. Typical Block Layout for an Engine-Pylon-Airframe Configuration.



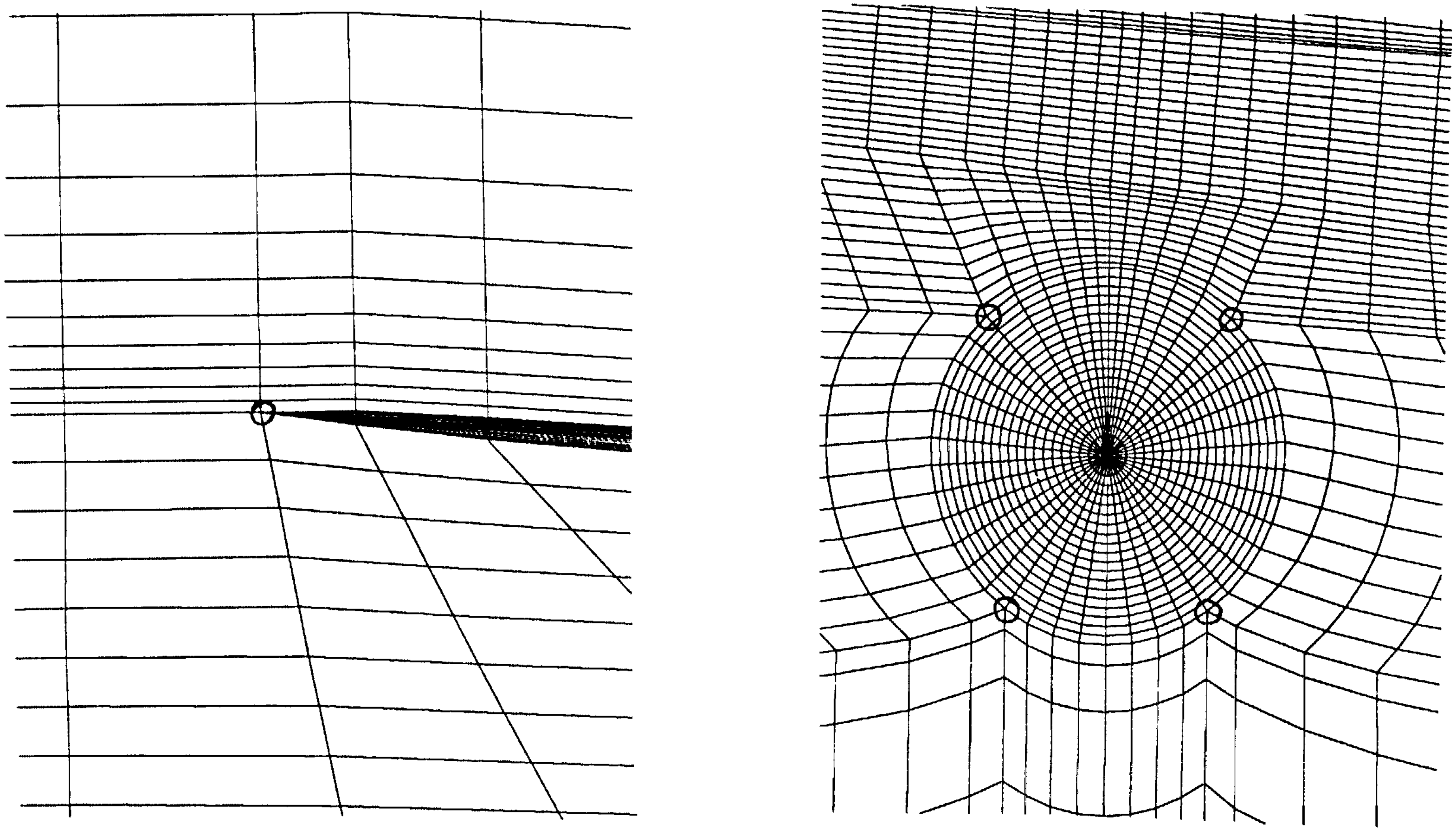
a. Surface Grid for the Airplane

Fig. 2. Surface Grid for an Engine-Pylon-Airframe Configuration.



b. Surface Grid Detail Near the Nacelle-Pylon-Wing Junction

Fig. 2. Concluded.



a. Collapsed Edge

b. Fictitious Corners

Fig. 3. Singular or Irregular Grids for a General Multiblock Solver.

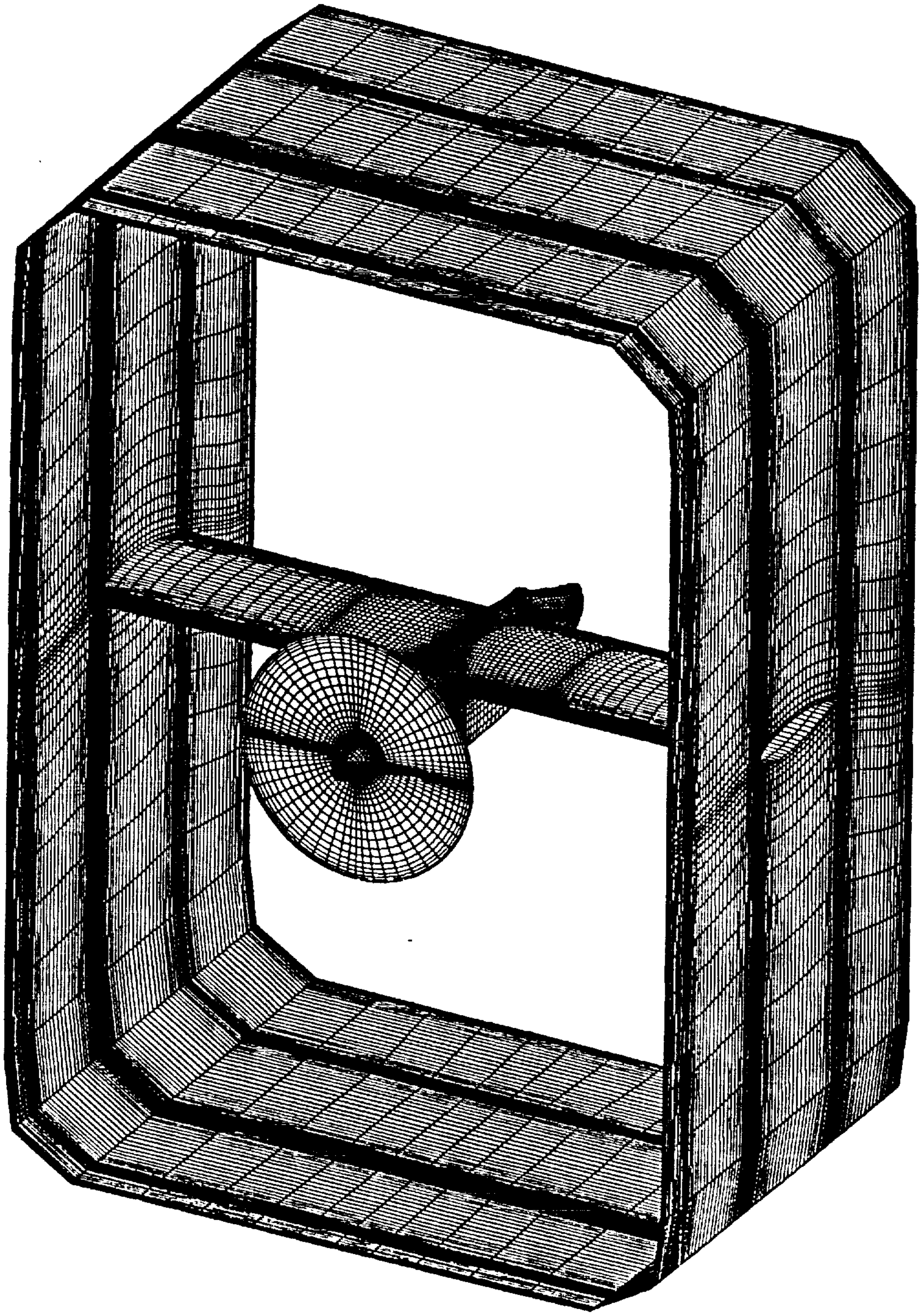
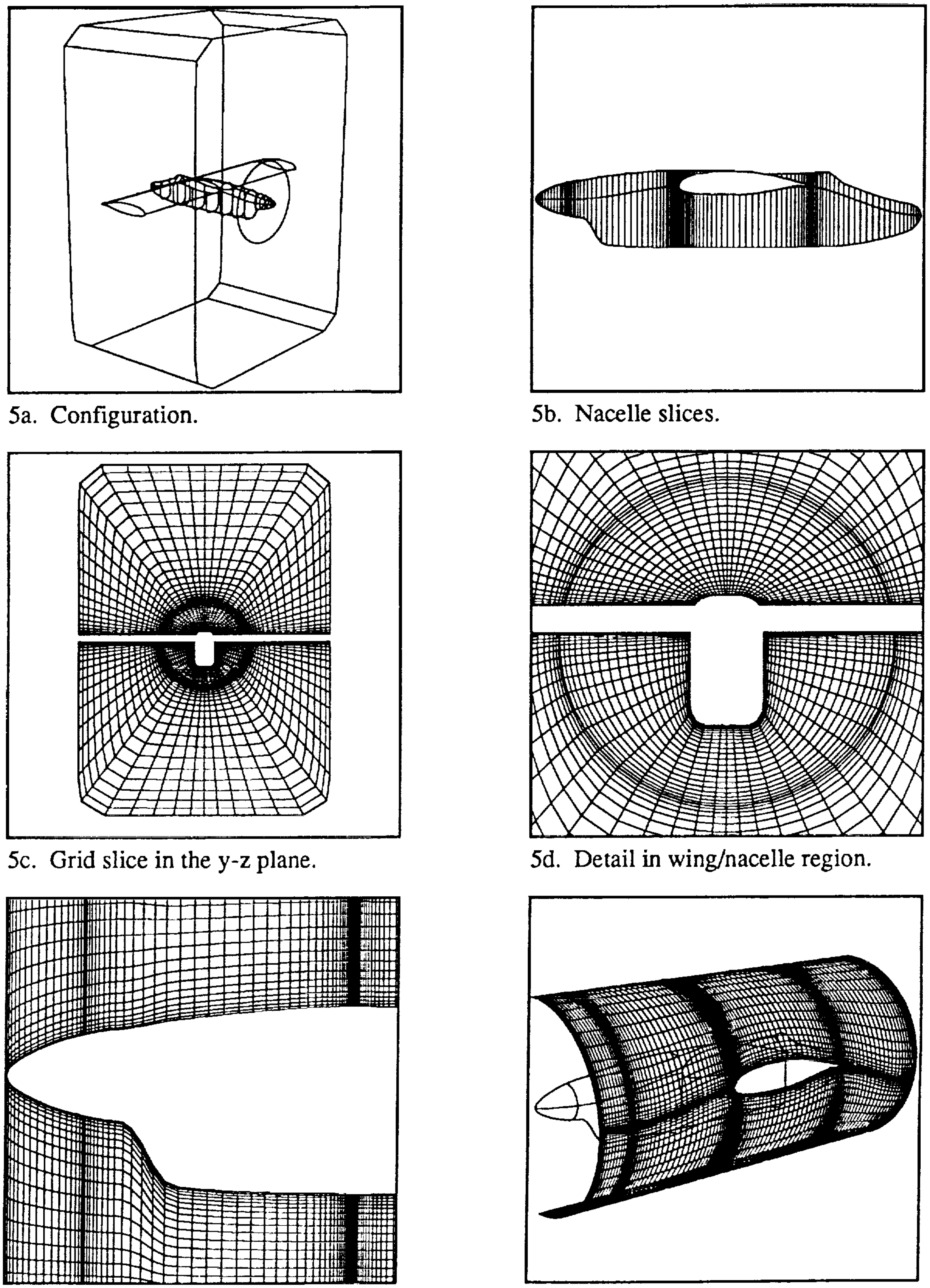


Fig. 4. Surface Grid for a Wing-Mounted Propfan in a Wind Tunnel.



5a. Configuration.

5b. Nacelle slices.

5c. Grid slice in the y-z plane.

5d. Detail in wing/nacelle region.

5e. Grid on nacelle symmetry plane.

5f. Grid surface at propeller tip.

Fig. 5. Configuration and Grid Slices.

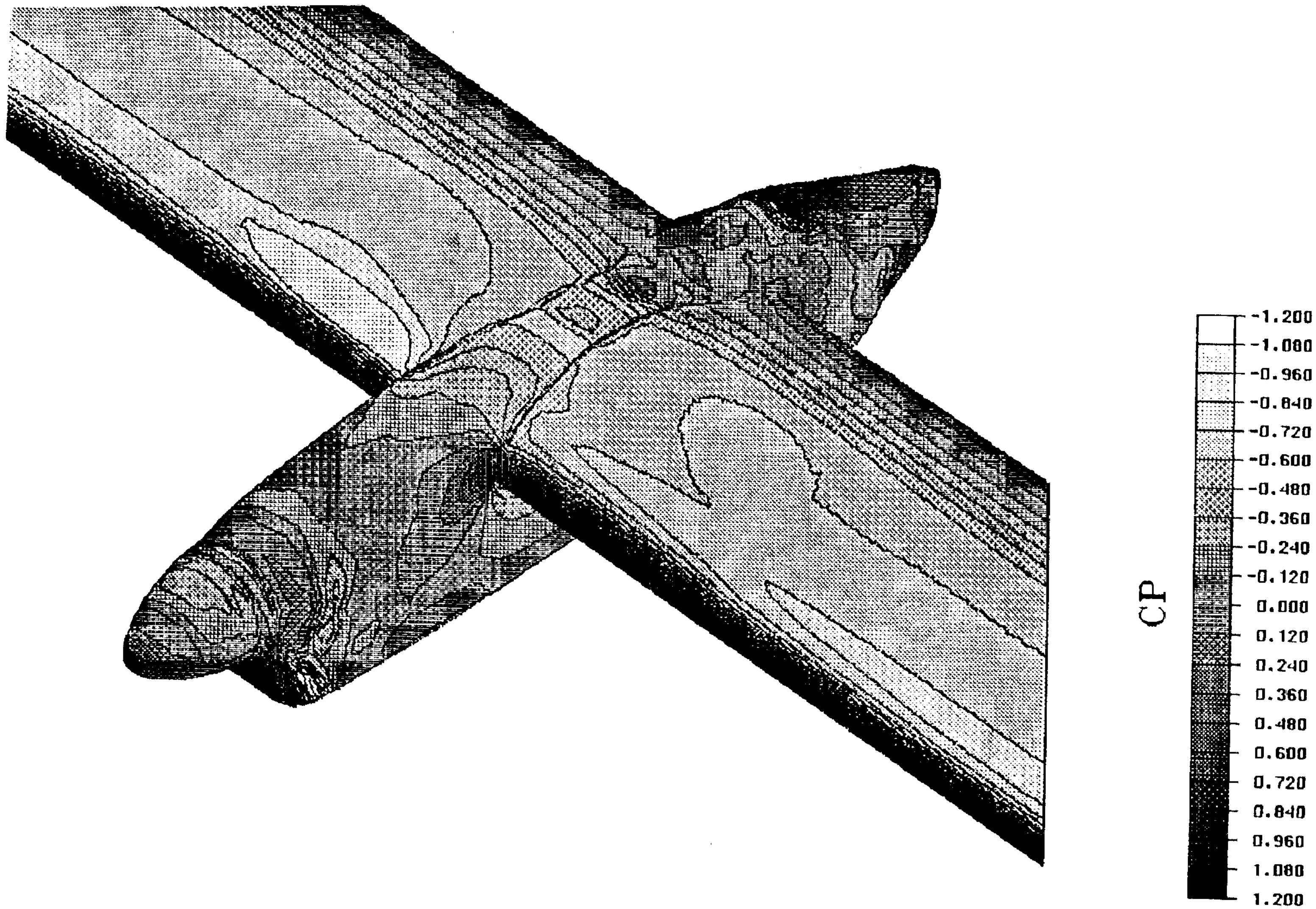


Fig. 6. GMBE Computed Isobars for Propeller Power-On, $M_\infty = .167, \alpha = 0.$

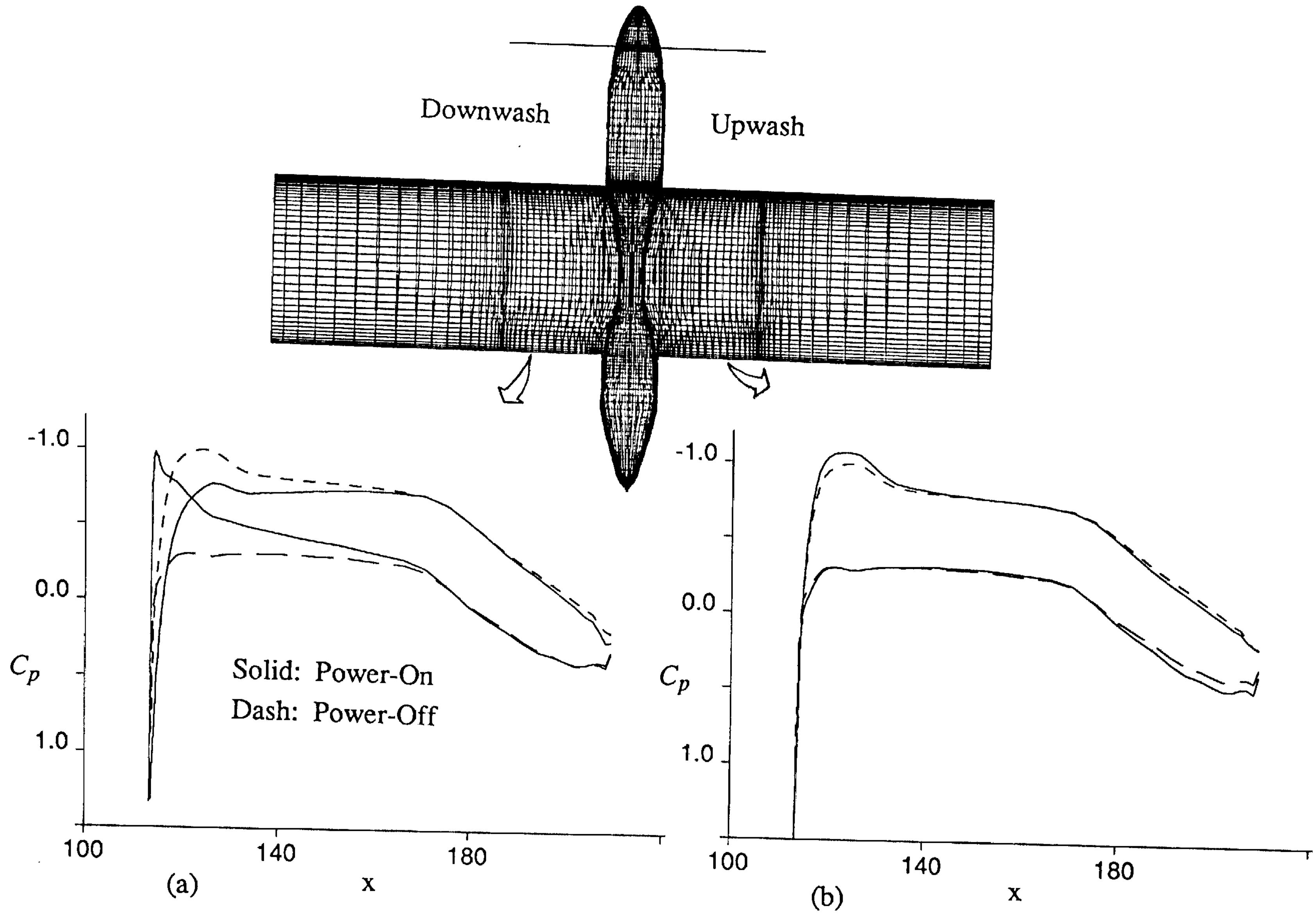


Fig. 7. Propeller Power Effects on Surface C_p, M_∞=0.167, α=0.0 deg.

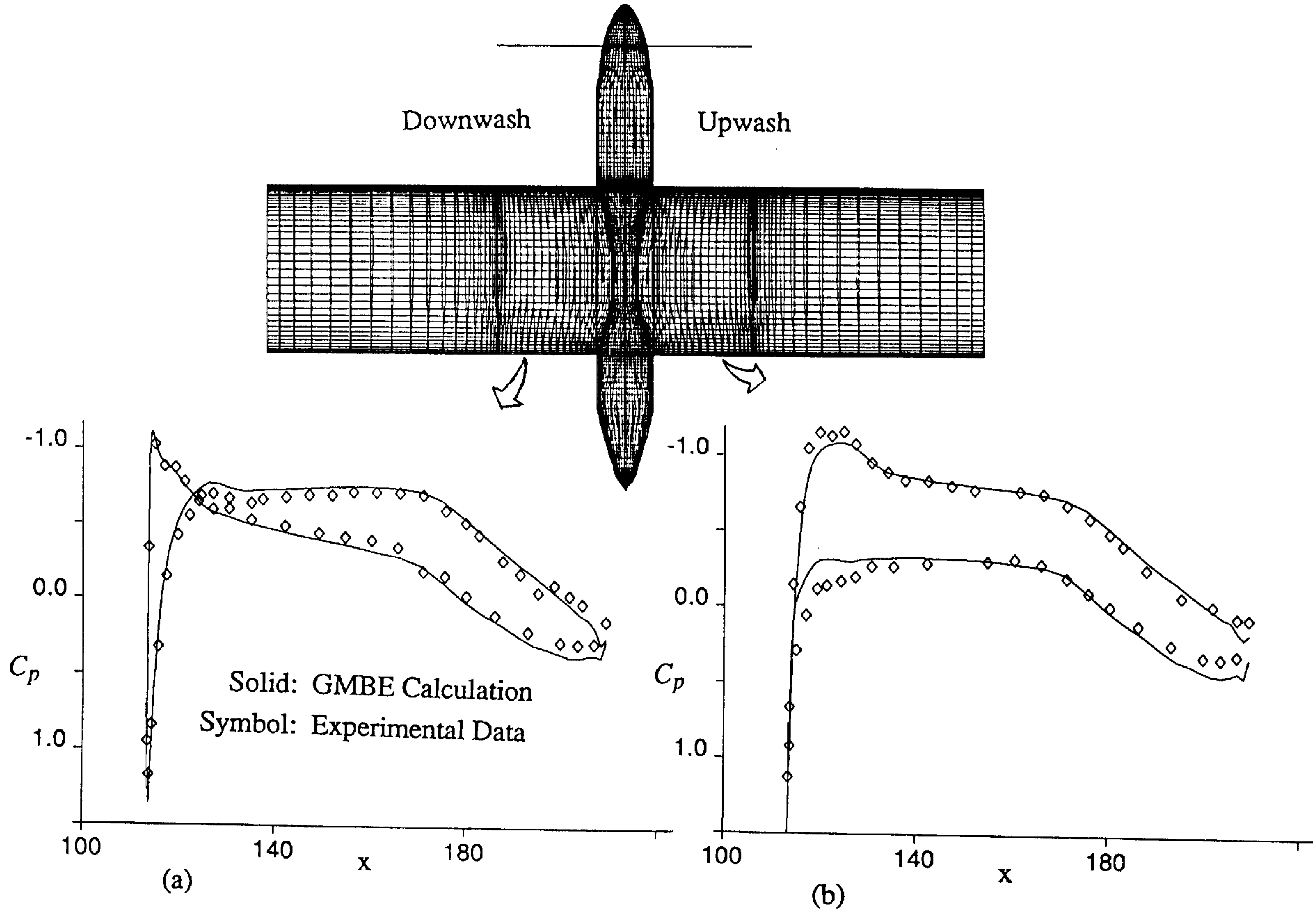
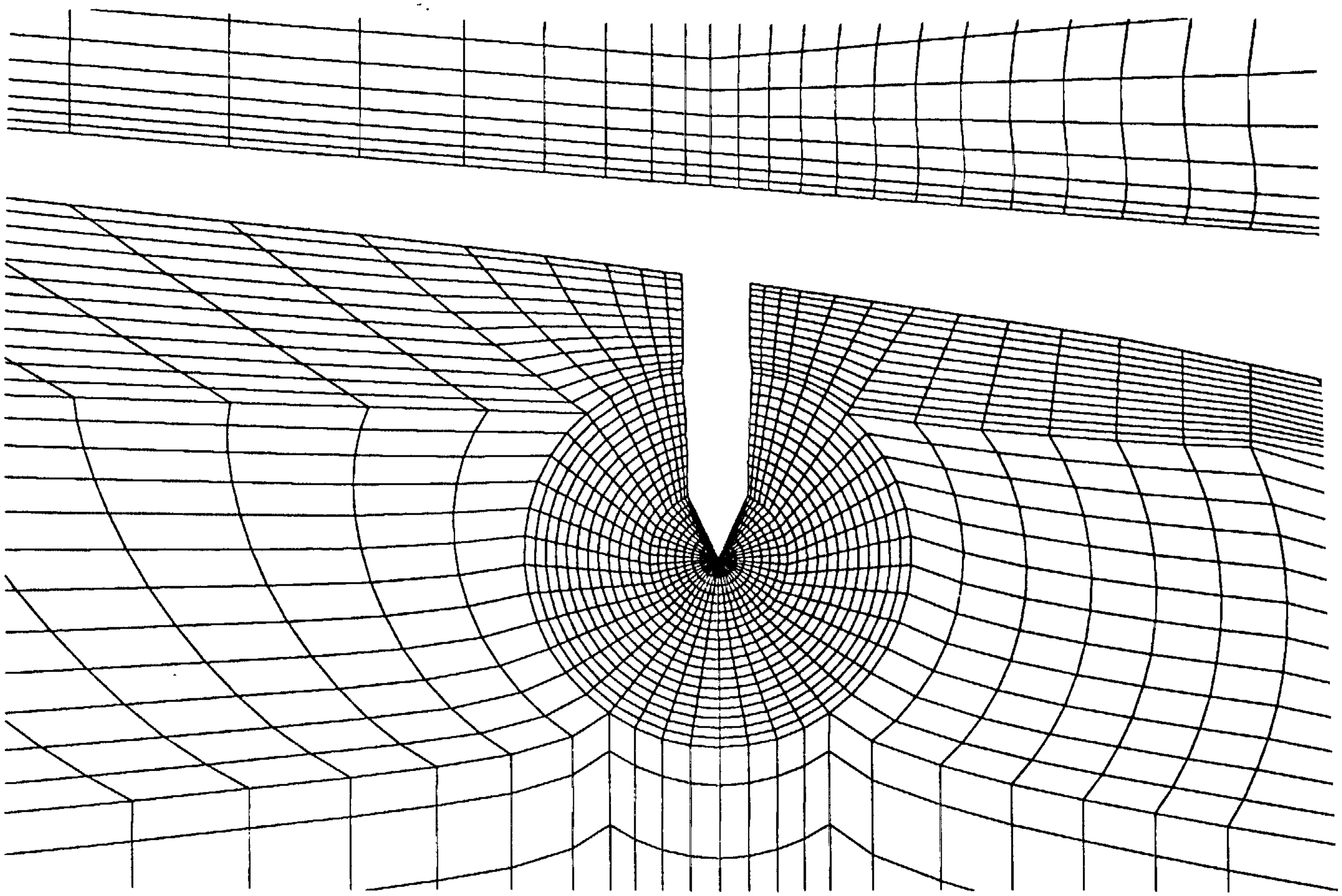
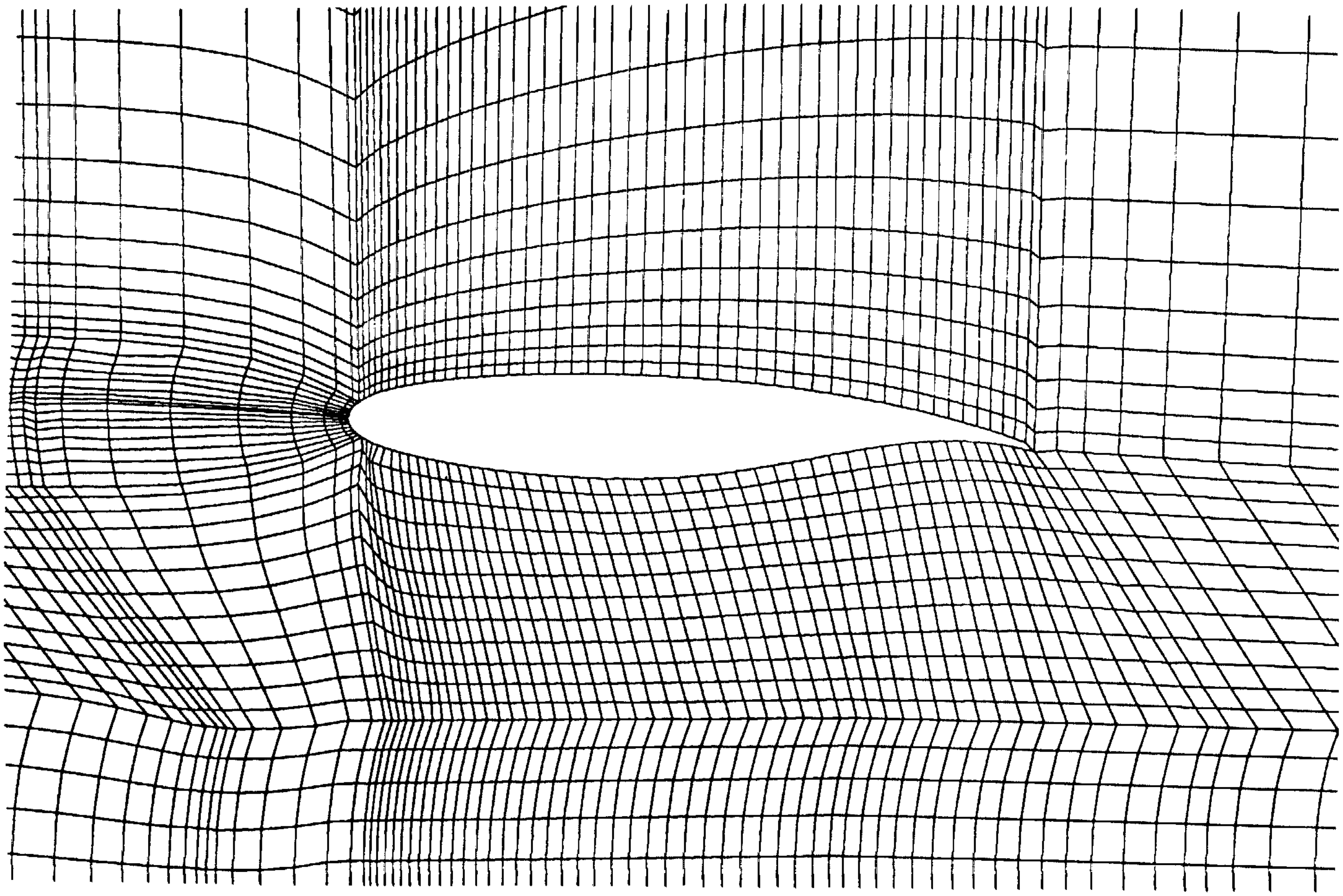


Fig. 8. Comparison of Experimental and Computed C_p Distributions for Propeller Power-On, $M_\infty = 0.167$, $\alpha = 0.0$ deg.



a. Cylindrical Grid in the Nacelle Exhaust Region

Fig. 9. Composite Grid System for an Engine-Pylon Installation.



b. H-Grid Around the Wing

Fig. 9. Concluded.

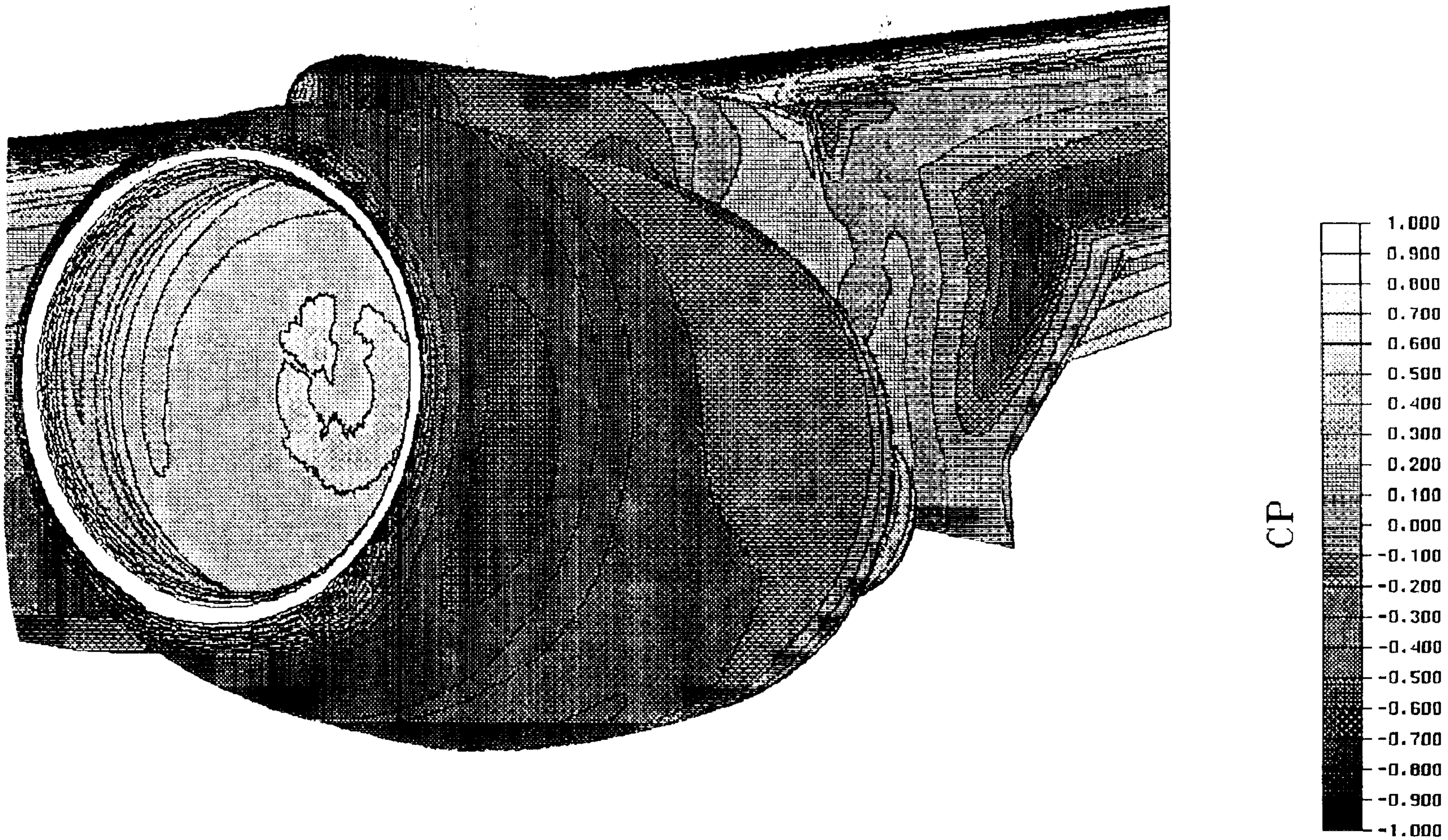


Fig. 10. GMBE computed Isobars for Flow Through Nacelle, $M_\infty = 0.77$, $\alpha = 0.50$ deg.

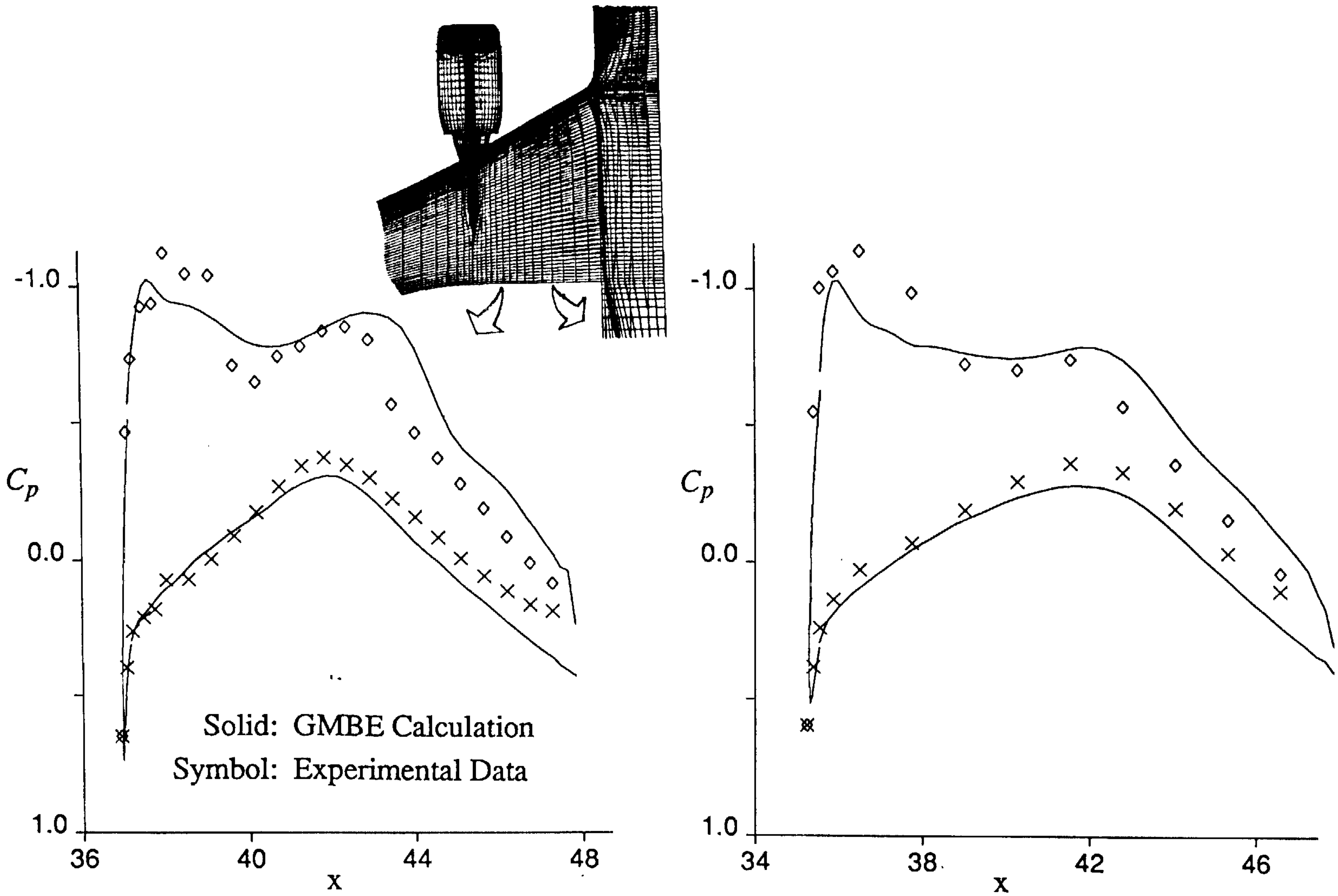


Fig. 11. Comparison of Experimental and Computed wing C_p Distributions for Flow-Through Nacelle, $M_\infty = 0.77$, $\alpha = 0.50$ deg.

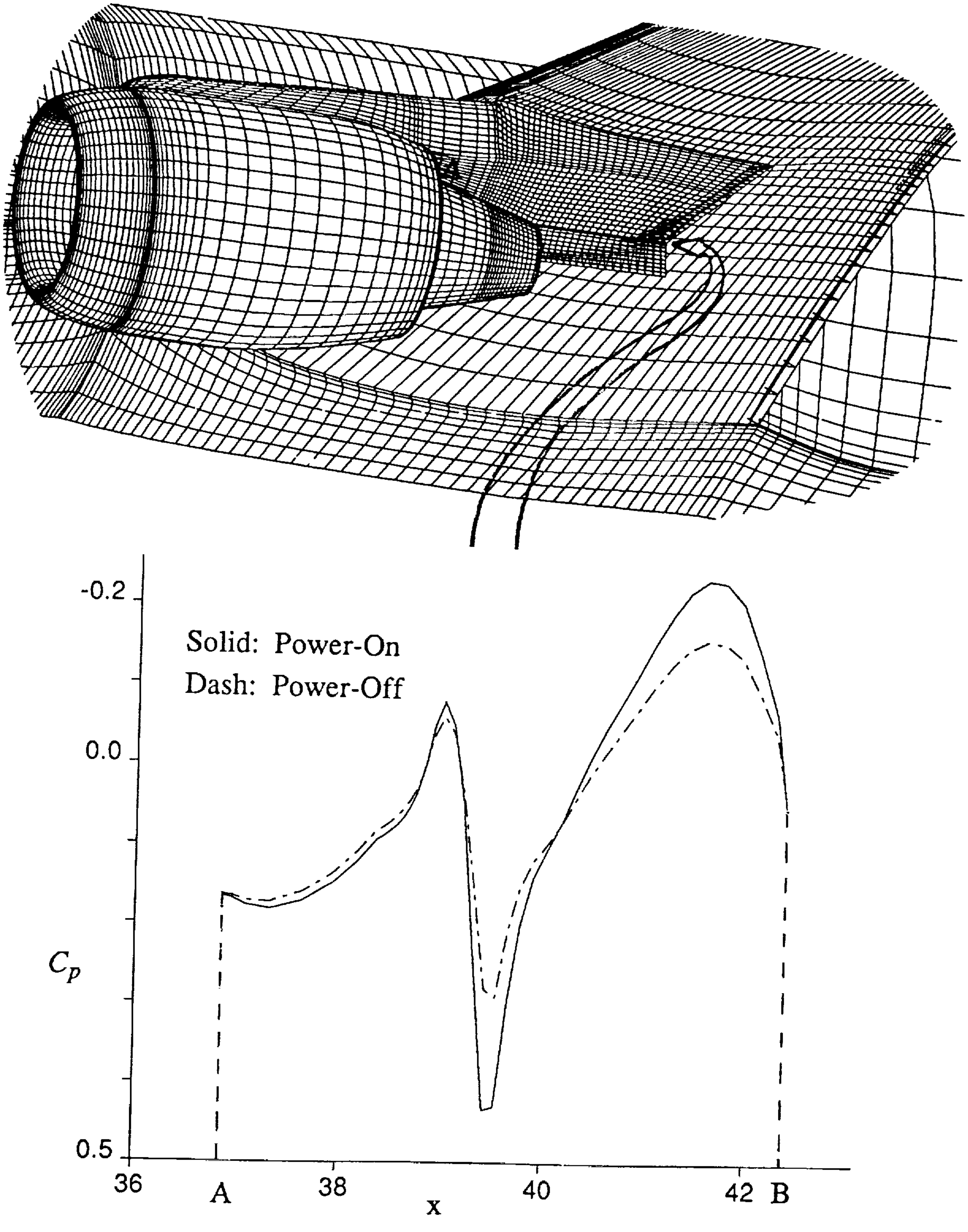


Fig. 12. Turbofan Engine Power Effect.

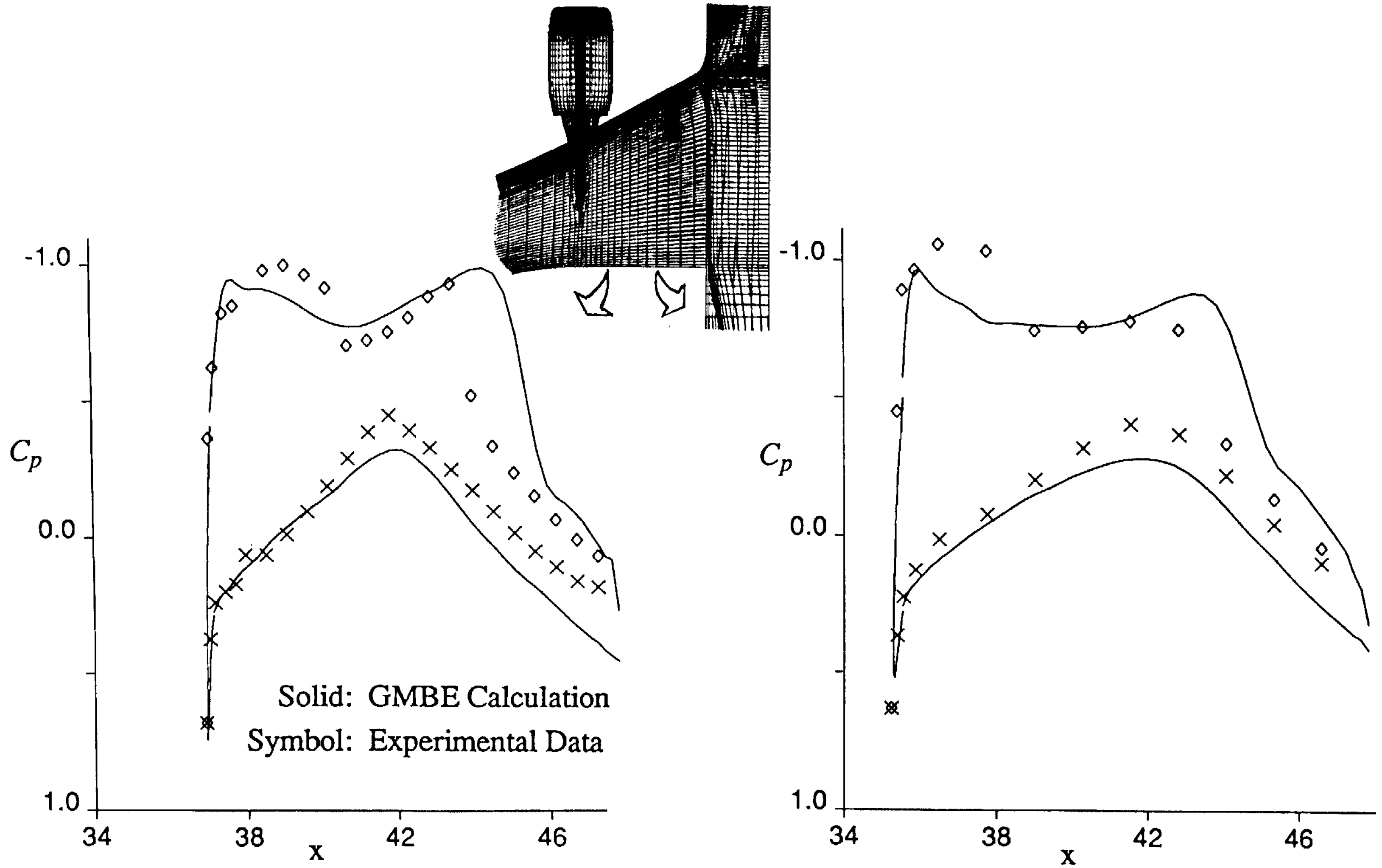


Fig. 13. Comparison of Experimental and Computed wing C_p Distributions for Flow-Through Nacelle, Off-Design Conditions, $M_\infty = 0.80$, $\alpha = 0.474$ deg.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA CR-187484, Volume I

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A GENERAL MULTIBLOCK EULER CODE FOR PROPULSION INTEGRATION, VOLUME I: THEORY DOCUMENT

5. Report Date
May 1991

6. Performing Organization Code

7. Author(s)
H. C. Chen, T. Y. Su, and T. J. Kao

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address
Boeing Commercial Airplane Group
P.O. Box 3707
Seattle, Wa. 98124-2207

11. Contract or Grant No.
NAS1-18703

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

14. Sponsoring Agency Code
535-03-10-01

15. Supplementary Notes
Langley Technical Monitor: Bobby Lee Berrier
Final Report

16. Abstract

A general multiblock Euler solver was developed for the analysis of flow fields over geometrically complex configurations either in free air or in a wind tunnel. In this approach, the external space around a complex configuration was divided into a number of topologically simple blocks, so that surface-fitted grids and an efficient flow solution algorithm could be easily applied in each block. The computational grid in each block is generated using a combination of algebraic and elliptic methods. A grid-generation/flow-solver interface program was developed to facilitate the establishment of block-to-block relations and the boundary conditions for each block. The flow solver utilizes a finite-volume formulation and an explicit time-stepping scheme to solve the Euler equations. A multiblock version of the multigrid method was developed to accelerate the convergence of the calculations. The generality of the method was demonstrated through the analysis of two complex configurations at various flow conditions. Results were compared to available test data. Two accompanying volumes, user manuals for the preparation of multi-block grids (Vol. II) and for the Euler flow solver (Vol. III), provide information on input data format and program execution.

17. Key Words (Suggested by Author(s))
Multiblock, Euler Code,
Propulsion Integration

18. Distribution Statement
Unclassified - Unlimited
Subject Catagory 02

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of pages
44

22. Price